PATENT

Docket: 00-088

Viterbi Decoder With Adaptive Traceback

By:

Feng Qian
25851 Majorca Way
Mission Viejo, CA92692
Citizenship: USA
949-582-2826

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to a method and apparatus for decoding convolutionally encoded signals. Such signals are commonly used in communications and recording systems that employ error correction to combat signal corruption.

Description of the Related Art

One increasingly popular redundant coding scheme used in wireless and other types of communications systems is convolutional coding, in which the coding of particular symbol depends upon the value of that symbol and the value of a certain number of symbols preceding and succeeding that particular symbol.

It would be desirable to improve upon conventional maximum likelihood sequence estimation ("MLSE") decoders such as Viterbi decoders, which are commonly used to decode convolutionally encoded data. If s_k is a vector that represents an actual transmitted sequence of symbols and r is a vector that represents the actual signals received by a mobile, a Viterbi decoder effectively tests all of the possible values of s_k and selects the s that maximizes the summation ("cross correlation") $\sum_n c[n]r[n]$ $s_k[n] = s_k^T C r$ where $c[n]$ defines the channel gain for a transmitted symbol for sample n (and thus C is a diagonal matrix with these values). (The s_k that maximizes

the cross correlation is the $\underline{s}_k$ that minimizes the "distance" between $\underline{r}$ and $\underline{s}_k$.) For more details regarding Viterbi decoders, see, for example, "Digital Communications", John G. Proakis (3d edition 1995).

5 The best $\underline{s}_k$ may be represented as a path through nodes in a diagram (known as a trellis), where lines ("transitions") between nodes in adjacent time steps represent whether an input symbol (information bit) was a 0 or a 1 (for binary coding). The nodes ("states") in vertical columns represent the values of prior input symbols. A "path" through the trellis therefore

10 represents a particular sequence of input symbols. Figure 1 shows an example of a trellis with only two paths shown. The time steps indicate a decoder trellis beginning at time t=0 to time t=4. At each time step, the decision units of the decoder contain the value of the cross correlation of the most likely path to the state of interest. Thus, the decoder trellis contains

15 multiple paths from an initial state to a given state several time steps later.

  However, only one path has the highest cross correlation and is the most likely path. The most likely path to a particular state within a given time step is found by starting at that particular state at that given time step

20 and tracing backward (a "traceback") along the chosen transitions. Information bits that correspond to the transitions along the path are the decoded data. An information bit equal to 0 is shown is a solid line; an information bit equal to 1 is shown as a dotted line. Thus, the path shown in Figure 1 ending at state A at time t=0 corresponds to an information bit

25 sequence equal to 0100.

A weight is computed for each state at each time, where the weight for a particular state at a particular time corresponds to the likelihood that the encoder was in that state at that time. The weight is equal to the weight of a previous state that transitioned to the current state plus a weight based upon the likelihood of the transition. For example, node 10 at time t=4 has a metric of four which equals the weight of node 12 at time t=3 plus the weight for the transition between those nodes. The node at any given time with the highest weight is the node that ends the most likely path and is therefore the node from which the traceback will occur.

To achieve good noise performance from the decoding process, the traceback length must generally be several times the constraint length (the number of input bits upon which an output depends; this is the length of a shift register that may be used to perform the encoding) of the code. In current code division multiple access ("CDMA"), a traceback length L = 5*K, where K is the constraint length, is frequently taken as the minimum acceptable traceback length.

For a decoder with M bits of memory, M-L valid bits can be obtained after each traceback. If an incoming data frame consists of N information bits (assumed to be greater than M), the number of L bit tracebacks is (N-K)/(M-L), which corresponds to a computation requirement of M*(N-K)/(M-L), where a computation is assumed to be a one bit traceback. (The one bit traceback is used simply as a "yardstick.") If M=L+1, this equation reduces

3

to (L+1)*(N-K), which corresponds to the smallest memory and the greatest computational overhead.

5   Instead of performing periodic, partial tracebacks, if the decoder memory is sufficiently large, an entire incoming data frame may be stored and a full traceback of length (N-K) performed. Thus the full traceback is less computationally intensive than the periodic traceback scheme by a factor of M/(M-L). However, larger memories (greater M) are costly.

10   It would be desirable to implement the partial traceback scheme, with its smaller memory requirement, while at the same time decreasing its computational overhead.

SUMMARY OF THE PRESENT INVENTION

These and other needs are met by the present invention, which provides a method and apparatus for decoding transmitted data that has been generated by encoding information data with a convolutional encoder that generates convolutional codes based on an input sequence of data, where the encoder has a constraint length K and a rate k/n.

According to an embodiment of the present invention, L-1 data bits of a traceback beginning at a time step T are stored, where L is the traceback length; these L-1 data bits are the data bits corresponding to the L-1 time steps backwards from time step T. The maximum likelihood encoder state for time T is also saved. (The L-th data bit is the desired data bit as in conventional convolutional decoders.) In a subsequent partial traceback, preferably beginning at time T+1 that ends at time step T, the maximum likelihood encoder state for time T determined from the partial traceback is compared with the stored encoder state for time T. If they correspond to the same encoder state, the L-1 stored data bits are designated as the last L-1 data bits of the current (partial) traceback.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the accompanying drawings in which:

Figure 1 is an example of a trellis diagram.

Figure 2 represents a digital communications system, in which a convolutional decoder including the present invention may be used.

Figure 3 is a flow chart that shows the operation of a possible embodiment of a convolutional decider constructed according to the teachings of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

The MLSE decoder that is described is a Viterbi decoder but the invention may be applied to any type of MLSE decoder.

5

General System Description

Figure 2 represents a digital communications system 140 comprising a discrete-time channel 142 interposed between an encoder 144 and a decoder

10   130. Discrete-time channel 142 comprises a modulator 146, a channel 148 and a demodulator 150. An interleaver 145 is interposed between the encoder 144 and the modulator 146. A deinterleaver 151 is interposed between the decoder 130 and the demodulator 150. Channel 148 may be a transmission channel or a storage medium being written to and read from.

15   Interleaver 145 receives a digital output signal from encoder 144 and interleaves this digital output signal over a certain time period, which is usually predetermined and known as a frame. Modulator 146 serves to translate the digital output signal from interleaver 145 into signals suitable for channel 148 and thereafter drives the signals across channel 148.

20

Channel 148 may suffer from interference that corrupts said signals, the interference possibly taking form in any combination of additive noise, cross channel interference, multi-path interference, and channel fading. Demodulator 150 serves to receive the signals from channel 148 while

25   minimizing the interference as much as is practical, and thereafter translate the signals into digital signals for input to deinterleaver 151, which

deinterleaves the digital signal and provides it to decoder 130. Discrete-time channel 142 can thus be viewed as a unit accepting digital input signals and producing possibly corrupted digital output signals although the present invention is not limited to noisy channels.

5

The decoder 130 has a memory 131.

Encoder 144 is a convolutional encoder which serves to add redundancy to input data signal 152. In particular, the encoder 144

10   comprises a shift register coupled to various arithmetic units (such as modulo 2 adders) that form n bits of output based on the bits in the shift register. For a simple shift register, where each shift of the register causes one bit to be shifted in and out of the register (i.e. each data bit is shifted to the next location in the register) each input bit corresponds to n bits of encoder 144

15   output and 1/n is known as the rate of the encoder 144. If the shift register has K elements, the K-1 most recent bits input into the encoder correspond to a state. K is known as the constraint length. For binary input data, there are therefore 2^{K-1} possible encoder states.

20   The added redundancy (n bits per information bit) allows for detection and correction of errors that may result from corruption of signals passing across discrete-time channel 142. The error detection and correction is performed by decoder 130.

Flow Chart of Operation

Figure 3 is a flow chart that details the traceback operation of the decoder 130. At block 200, the decoder 130 receives input data to be decoded. At block 210, the decoder 130 computes the current branch metric. This block is known in the art will not be further discussed. At block 220, the decoder 130 selects the surviving state. This block is well known in the art and will not be further discussed. At block 230, the decoder 130 checks whether L+K decoding cycles (i.e. one decoding cycle per time step) have occurred, where L is the traceback length and K is the constraint length. If not, control passes back to block 210. If so, control passes to block 240, where the decoder 130 stores the starting state of the traceback in the memory 131 (e.g., with reference to Figure 1, the state A at time t=4).

At block 250, the encoder determines whether the traceback to be performed is the first traceback of the current data frame. If so, control passes to block 260, where a full traceback is performed. For each time step transition within the traceback, the data bit corresponding to the transition between the current state (e.g. the encoder state at time step t=4) and the next most recent state (e.g. the state at time step t=3) is stored in the memory 131. Control passes back to block 210. Otherwise, if at least one traceback has already occurred for the data frame, control passes to block 270.

In block 270, the decoder 130 performs a partial traceback; in the preferred embodiment, the partial traceback traces back one time step. Control passes to block 280, which compares the finishing state of the partial

9

traceback (e.g. the state at time step t=4 for a traceback beginning at time t=5) to the initial state of the most previous traceback (e.g. the traceback that began at time step t=4). If these quantities are not equal, control passes back to block 260, and a "normal" traceback is performed (beginning from the end

5   of the partial traceback).

If these quantities are equal, control passes to block 290, where the L-1 bits stored from the previous traceback are designated as the L-1 final bits of the current traceback. In other words, the current traceback bits are taken

10   to be the concatenation of (a) the bit corresponding to the first encoder transition of the current traceback; with (b) the L-1 final bits of the previous traceback. The last bit stored from the previous traceback (i.e. the bit corresponding to the transition from the third to last encoder state to the second to last encoder state of the previous traceback) is designated as the

15   desired data bit for the traceback. (The transition from the third to last state to the second to last state of the previous traceback corresponds to the transition from the second to last state to the last state of the current traceback.)

20   Conclusion

Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.